Project Report

Project Name: **SMART SOLUTIONS FOR RAILWAYS**

Team ID: **PNT2022TMID37262**

Team: **GOKUL M - TEAM LEAD**

KASINATHAN M

BHUVANESHWAR S

LAKSHMANA SHANKAR B

1. INTRODUCTION

1.1 Project Overview

As trains are one of the most preferred modes of transportation among middle class and impoverished people as it attracts for its amenities. Simultaneously there is an increase at risk from thefts and accidents like chain snatching, derailment, fire accident. In order to avoid or in better words to stop all such brutality we came up with a solution by providing an application which can be accessed by the user after booking their tickets. With a single click this app addresses issues by sending a text message to TC and RPF as an alert. In our project we use Node-Red service, app-development, IBM cloud platform to store passenger data.

1.2 Purpose

The purpose of this project is to report and get relived from the issues related to trains.

2. LITERATURE SURVEY

2.1 ABSTRACT:

Indian Railways is the largest railway network in Asia and additionally world's second largest network operated underneath a single management. Due to its large size it is difficult to monitor the cracks in tracks manually. This paper deals with this problem and detects cracks in tracks with the help of ultrasonic sensor attached to moving assembly with help of stepper motor. Ultrasonic sensor allows the device to moves back and forth across the track and if there is any fault, it gives information to the cloud server through which railway department is informed on time about cracks and many lives can be saved. This is the application of IoT, due to this it is cost effective system. This effective methodology of continuous observation and assessment of rail tracks might facilitate to stop accidents. This methodology endlessly monitors the rail stress, evaluate the results and provide the rail break alerts such as potential buckling conditions, bending of rails and wheel impact load detection to the concerned authorities.

2.2 EXISTING SYSTEM:

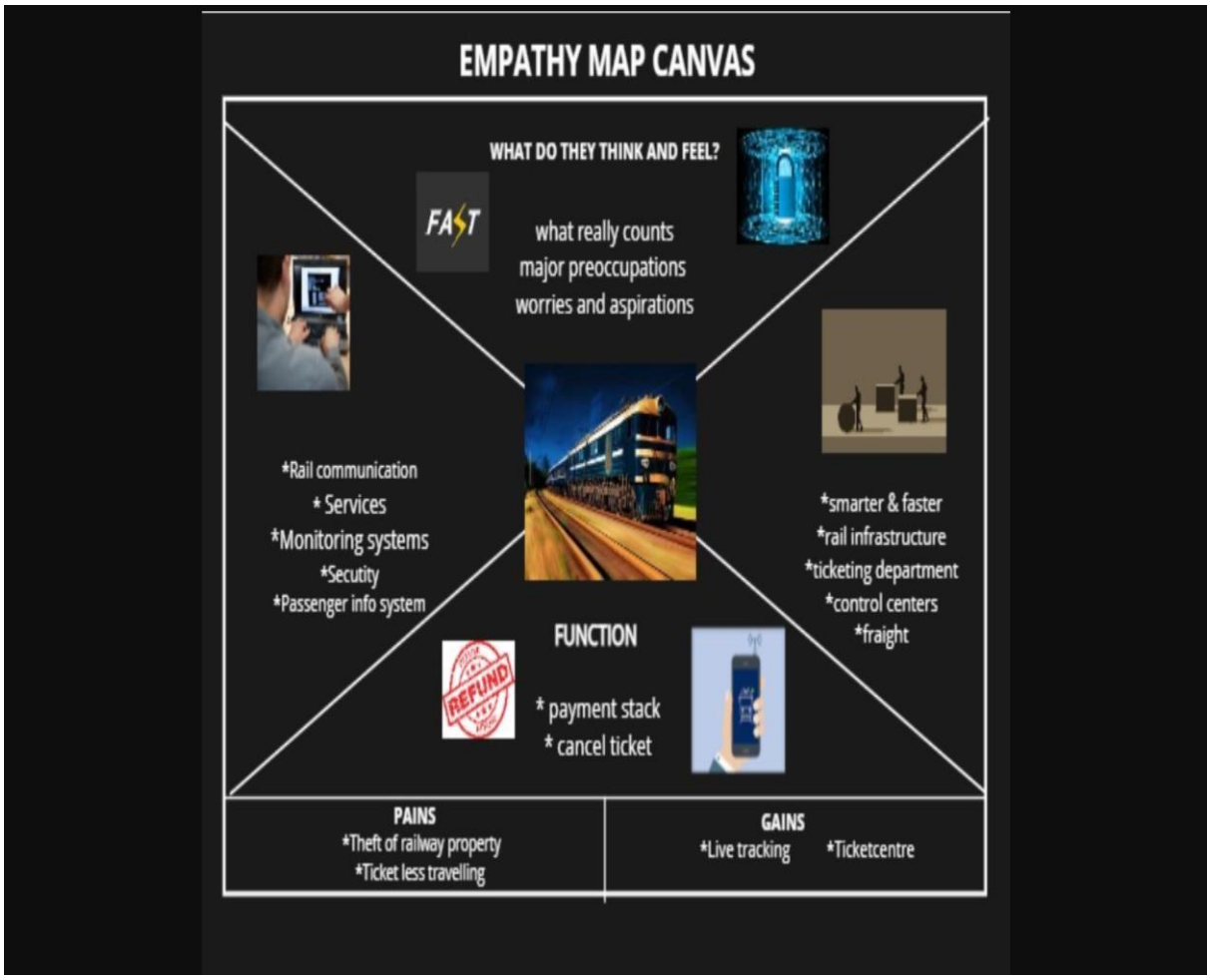
Existing train tracks are manually researched. LED (Light Emitting Diode) and LDR (Light Dependent Resister) sensors cannot be implemented on the block of the tracks]. The input image processing is a clamorous system with high cost and does not give the exact result. The Automated Visual Test Method is a complicated method as the video color inspection is implemented to examine the cracks in rail track which does not give accurate result in bad weather. This traditional system delays transfer of information. Srivastava et al., (2017) proposed a moving gadget to detect the cracks with the help of an array of IR sensors to identify the actual position of the cracks as well as notify to nearest railway station. Mishra et al., (2019) developed a system to track the cracks with the help of Arduino mega power using solar energy and laser. A GSM along with a GPS module was implemented to get the actual location of the faulty tracks to inform the authorities using SMS via a link to find actual location on Google Maps. Rizvi Aliza Raza presented a prototype in that is capable of capturing photos of the track and compare it with the old database and sends a message to the authorities regarding the crack detected. The detailed analysis of traditional railway track fault detection techniques is explained in table.

2.3 Problem Statement Definition :

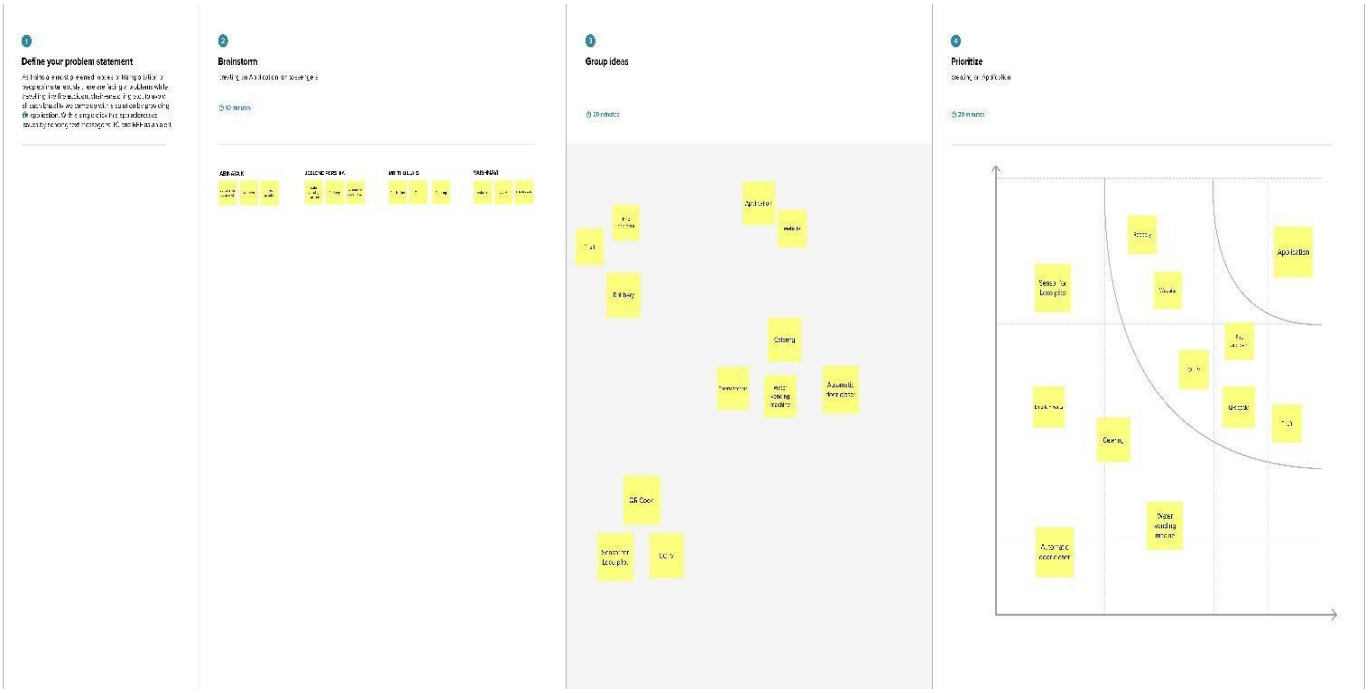
Smart Solutions for railways are designed to reduce the work load of the user and the use of paper.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

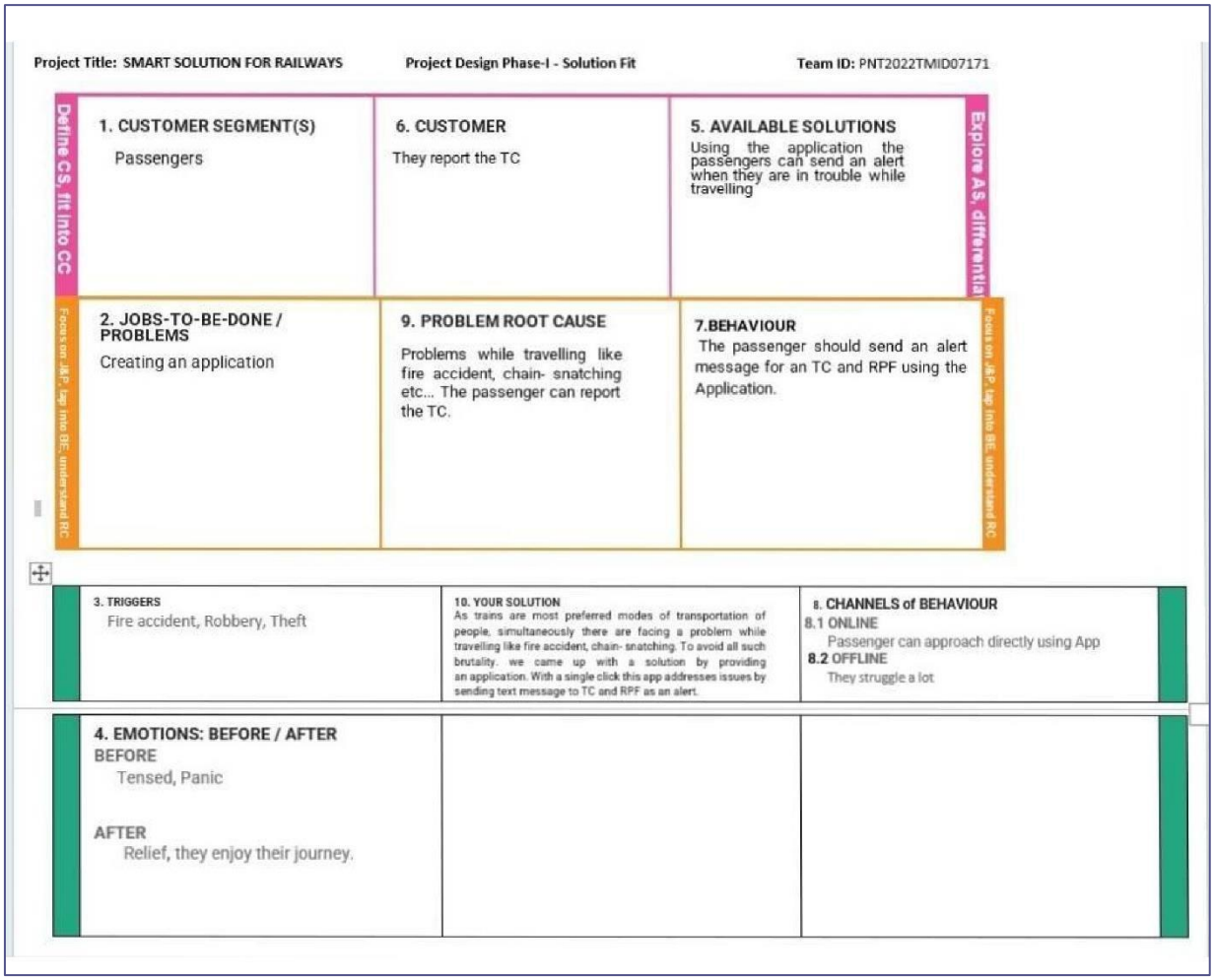


3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Problems in the railways like robbery, fire accidents etc..
2.	Idea / Solution description	Developing an app for the passengers.
3.	Novelty / Uniqueness	The passengers can send an alert to the respective officials during the travel time through the app when they are in trouble so that they can easily solve it.
4.	Social Impact / Customer Satisfaction	Usage of this app can be a great relief to the passengers, so that they can travel without any fear.
5.	Business Model (Revenue Model)	5000

6.	Scalability of the Solution	This solution will be useful for passengers while travelling. They can use the app between the time of their travel. The users will fell more secured, in-case of an emergency by simply clicking on a button the alert signal will be sent to the respective officials and the corresponding measures will be taken.

3.4 Problem Solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

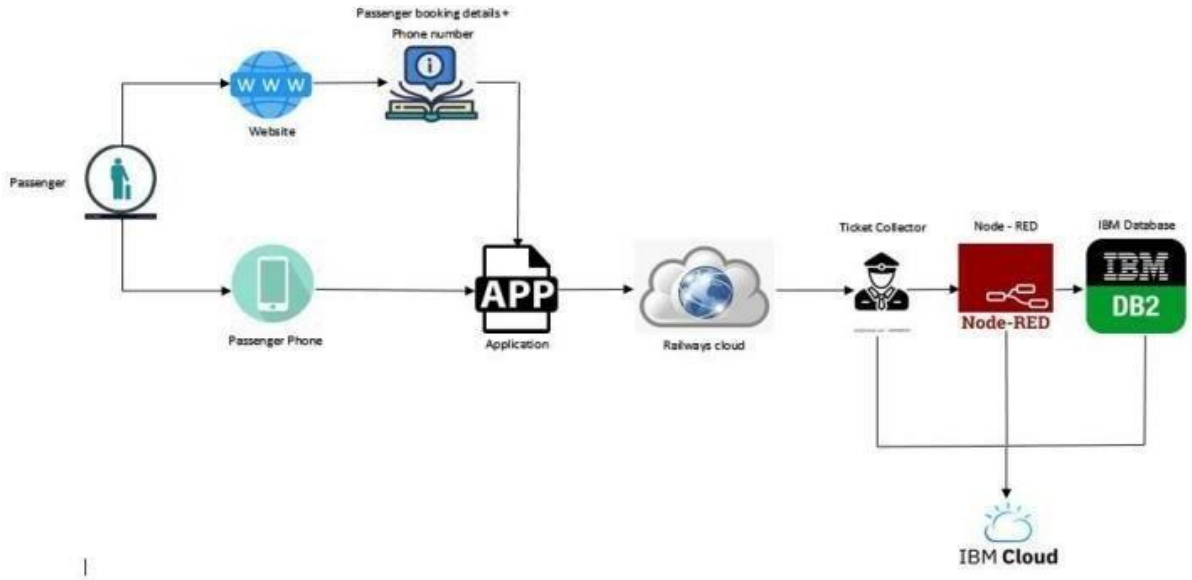
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User registration	Before the user registration there will be language selector .All the language is applicable. When user enter in to the application they can see the page of showing enter the email ,mobile number and name. After that in screen it shows the verification code is sent through the email id.
FR-2	User confirmation	The verification code is entered in to the app application. After finishing that home page is opened.
FR-3	Process of booking	When the home page is opened there will be a from and to option. We must enter the details then after that we can able to see the number of trains availability and seats availability. We can select the particular train and particular seats which we need and click the confirm option.
FR-4	Process confirmationn	After all the QR code will be send through the SMS and email id. QR code will be shown to the ticket collector when the QR code is scanned booking details will be shown .

4.2 Non-Functional requirement

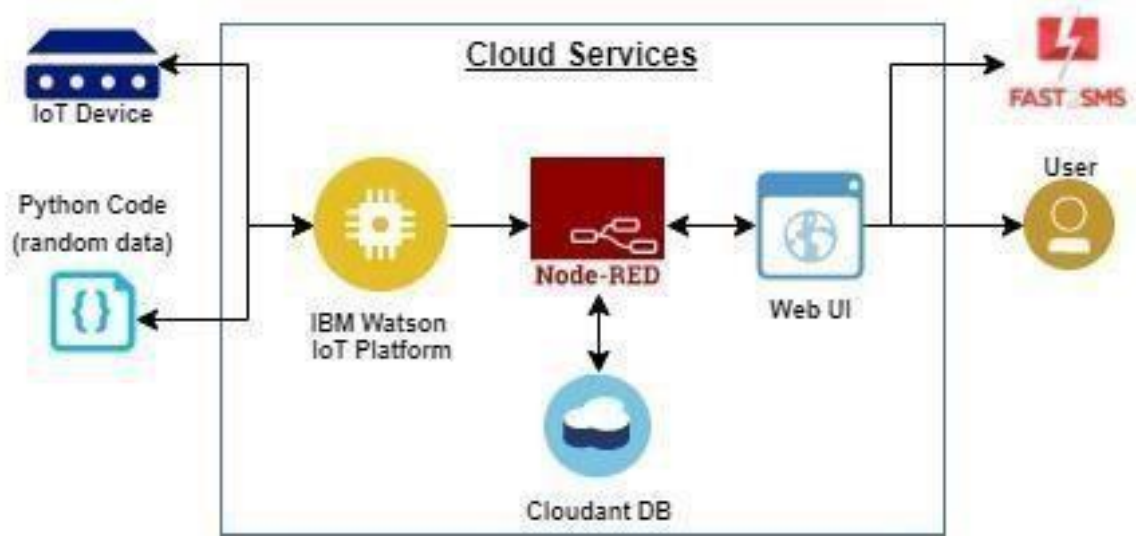
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	• The app can be used during the travelling time • Easy and simple • Efficiency is high
NFR-2	**Security**	By clicking on the icon, the alert will be given to the respective officials
NFR-3	**Reliability**	Highly reliable to use
NFR-4	**Performance**	Low error rate
NFR-5	**Availability**	Free source
NFR-6	**Scalability**	It is scalable enough to support many users at the same time

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution Architecture



5.3 User Stories :

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
PASSENGER (Mobile user)	Booking registration	USN-1	As a passenger, I book the ticket for the journey by entering my personal information.	I can access the web link to install the application.	High	Sprint-1
	Confirmation	USN-2	As a passenger, I will receive confirmation of the booking once I have registered for the application	I can receive confirmation email & click confirm.	High	Sprint-1
	Application registration	USN-3	As a passenger, I can register for the application through the weblink.	I can register & access the application through google login.	Low	Sprint-2
	Application access	USN-4	As a passenger, I can access the application during my travel for resolving my issues.		Medium	Sprint-1

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

STEP 1	Identify the problem
STEP 2	Prepare an abstract, problem statement
STEP 3	List required objects needed
STEP 4	Create a code and run it
STEP 5	Make a prototype
STEP 6	Test with the created code and check the designed prototype is working
STEP 7	Solution for the problem is found

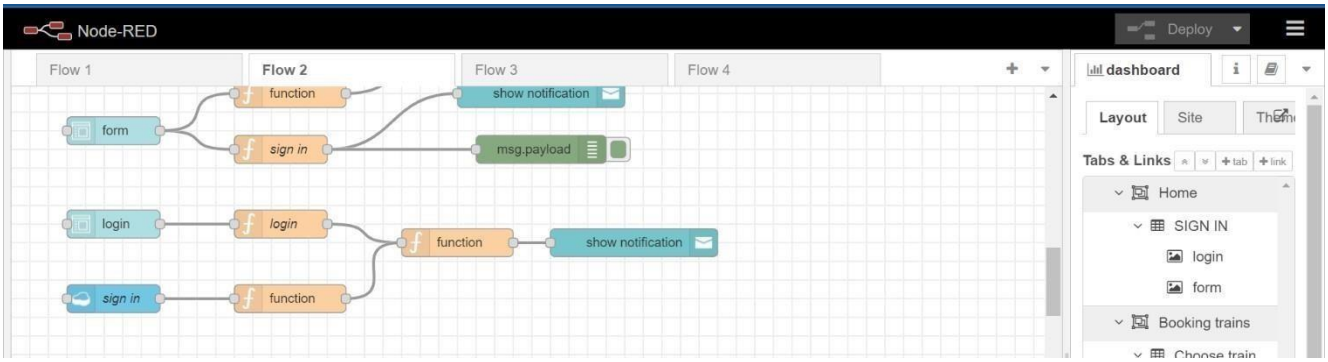
6.2 Reports from SPRINT

SPRINT 1

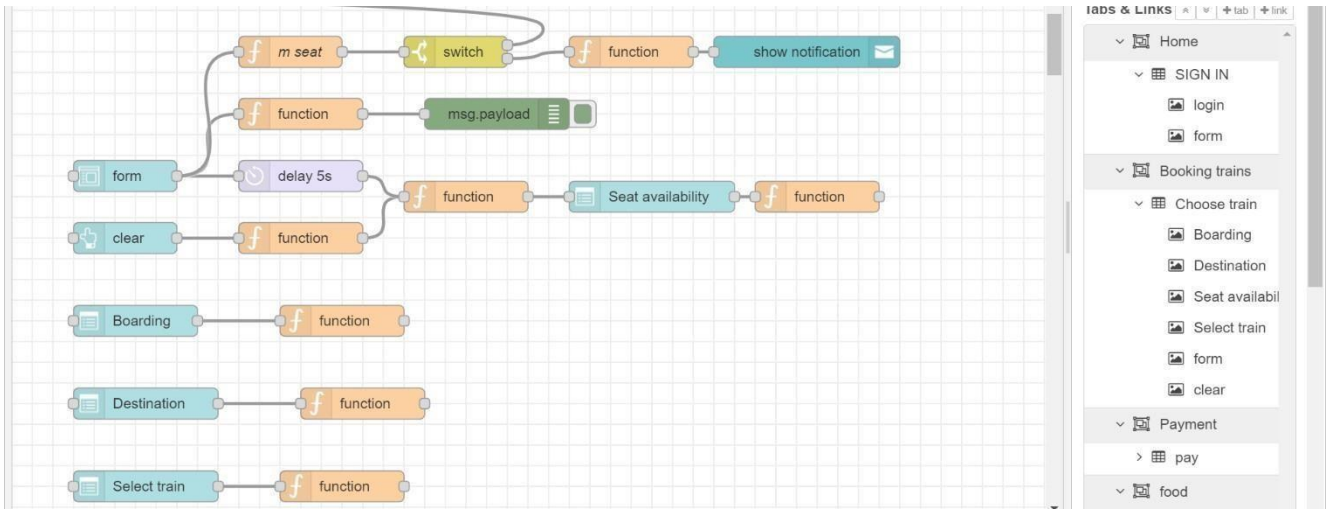
PROCEDURE:

Step1: Develop node red application
Step2: Install the required nodes from manage palette option
Step3: Connect the node flow
Step4: Deploy the flow

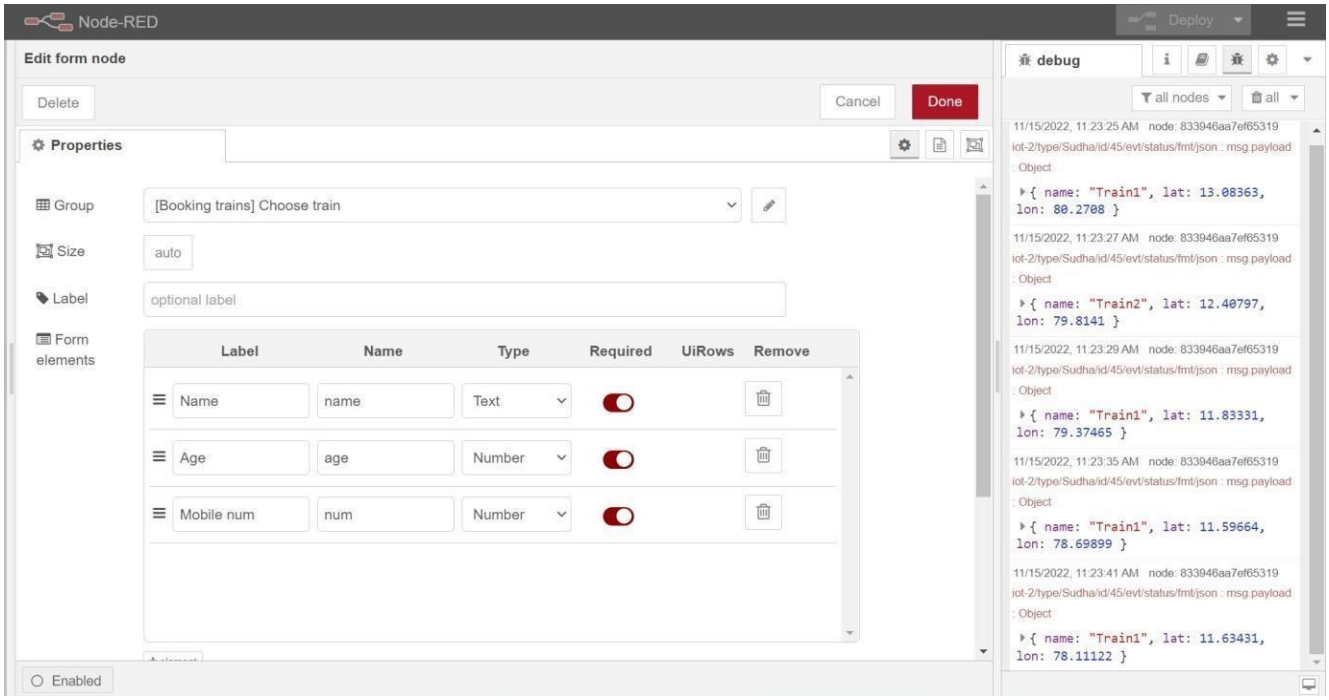
WEB APPLICATION :



NODES TO BOOK TRAIN:



FORM DETAILS:

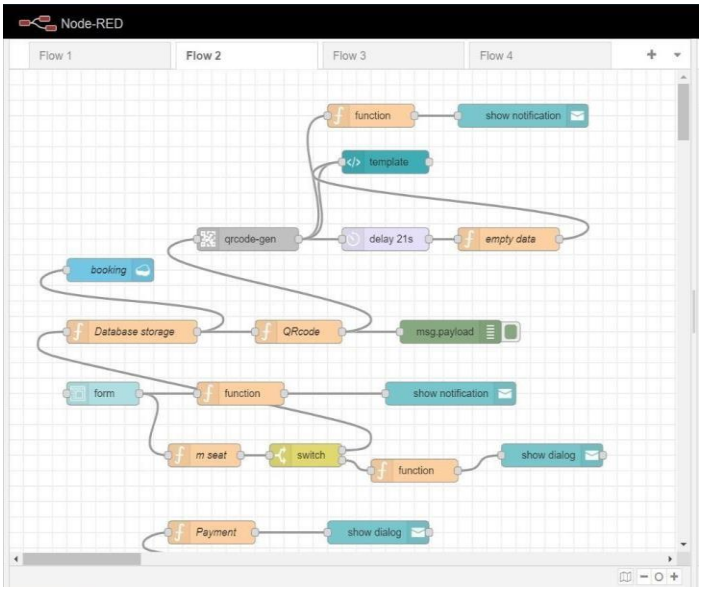


SPRINT 2

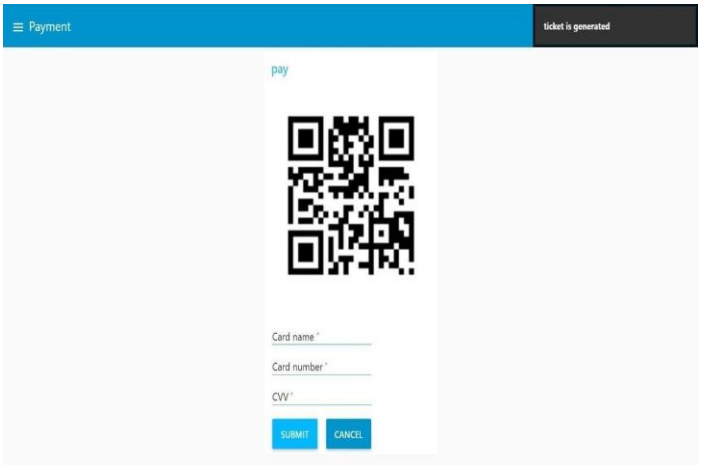
PROCEDURE:

Step1: Develop node red web application for train ticket booking
Step2: Copy the node red link and add /ui to the same link and browse it
Step3: Fill the details
Step4: Click on submit
Step5: QR code will be submitted
Step6:Ticket is generated

NODE RED FLOW CONNECTION:



QRCODE GENERATION:



SPRINT 3

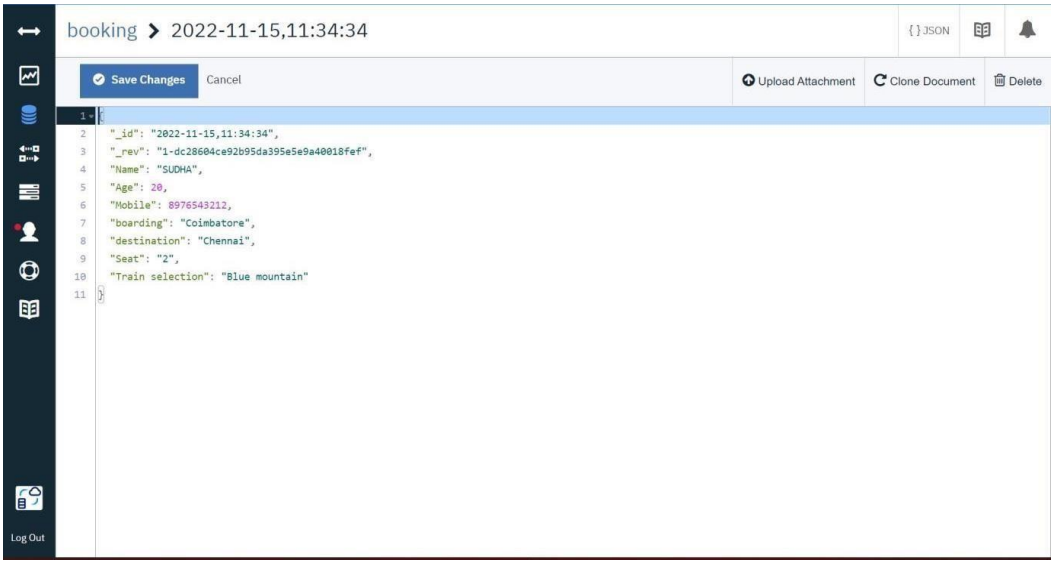
Step1: Develop a python script to scan the QR code

Step2: Connect the python code to IBM Cloudant using the credentials

Step3: Run the program

QR CODE DETAILS:

DATA STORED IN CLOUDANT:



SPRINT 4

Step1: Develop a node red application for GPS

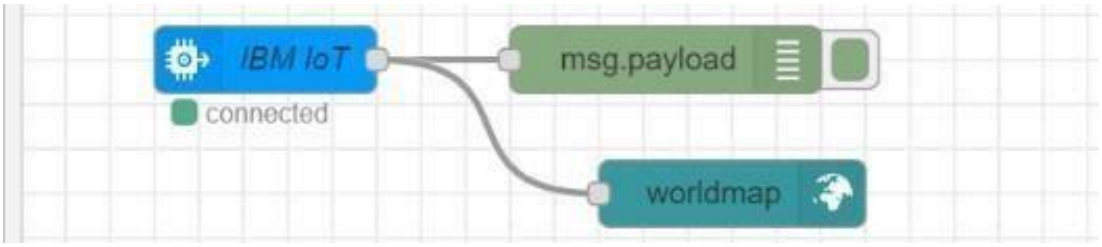
Step2: Develop a python code for GPS

Step3: Run the program

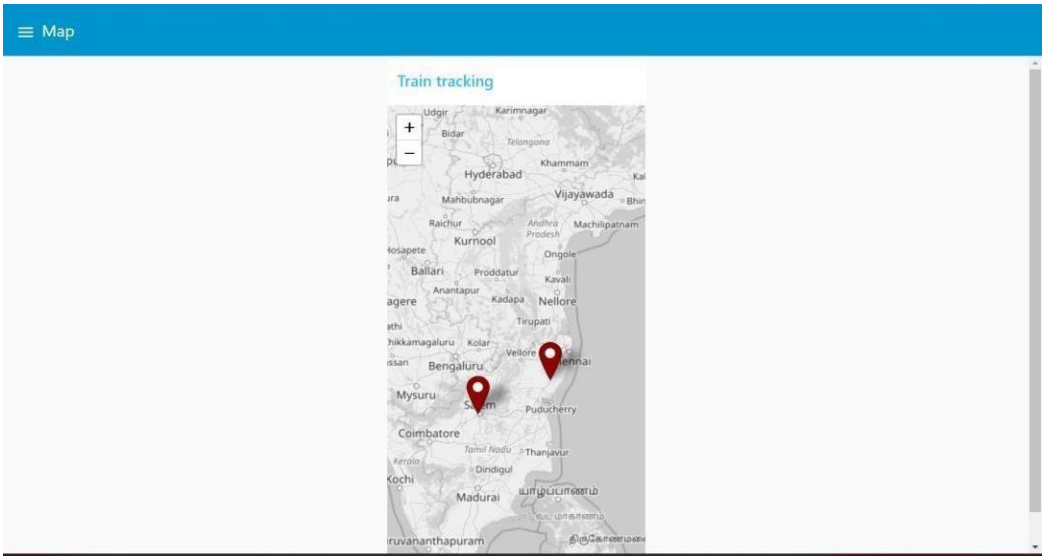
Step4: Train location will be displayed

Step5:Create a node red for wakeup call and E-catering serice

NODE RED FLOW:



TRAIN TRACKING :



7. CODING & SOLUTIONING

7.1 Feature 1

- IoT device
- IBM Watson Platform
- Node red
- Cloudant DB
- Web UI
- Python code

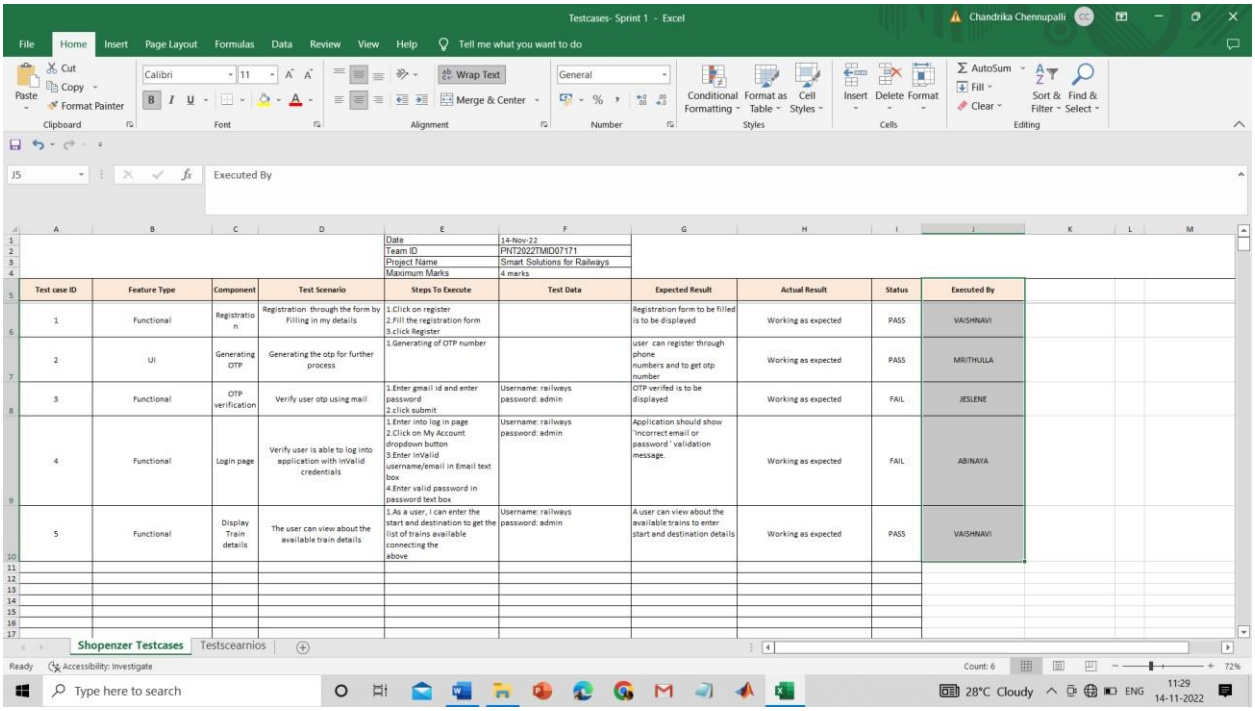
7.2 Feature 2

- Login
- Verification
- Ticket Booking
- Adding rating
- QR code
- GPS tracking

8. TESTING AND RESULTS

8.1 Test Cases

Test case 1

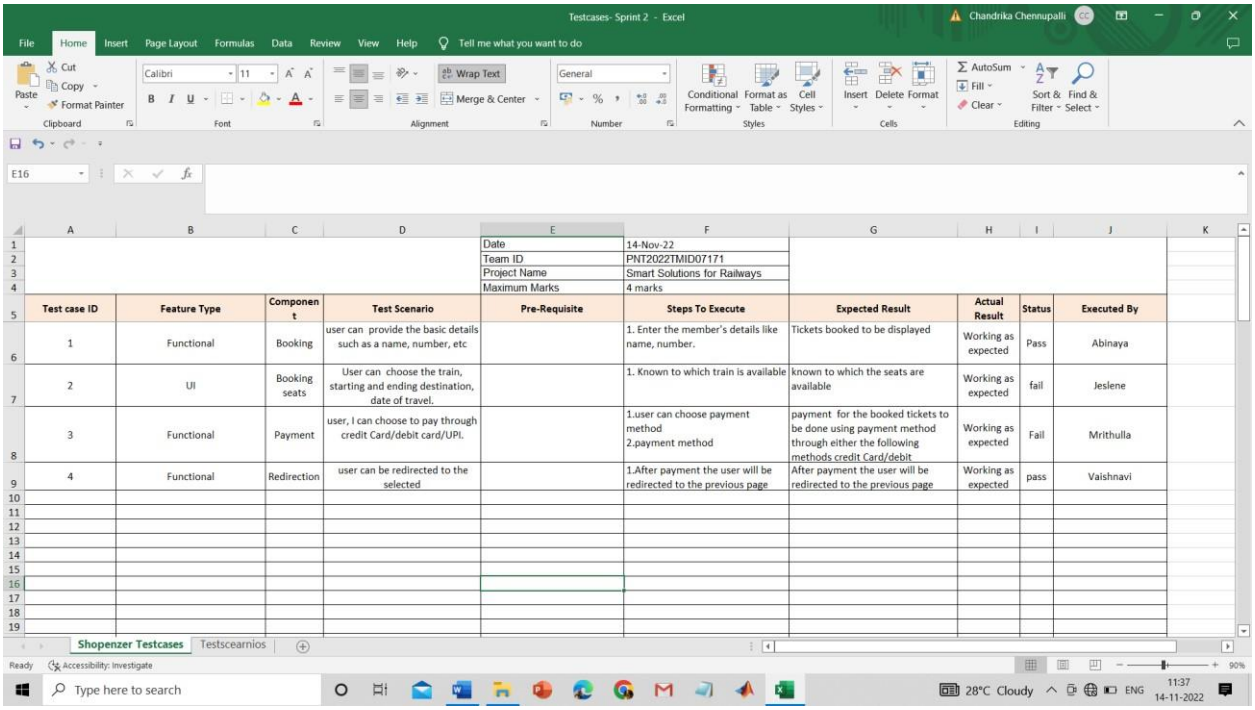


Testcases- Sprint 1 - Excel

				Date	14-Nov-22				
				Team ID	PNT2022TMID07171				
				Project Name	Smart Solutions for Railways				
				Maximum Marks	4 marks				
Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Executed By
1	Functional	Registration	Registration through the form by Filling in my details	1.Click on register 2.Fill the registration form 3.click Register		Registration form to be filled is to be displayed	Working as expected	PASS	VAISHNAVI
2	UI	Generating OTP	Generating the otp for further process	1.Generating of OTP number		user can register through phone numbers and to get otp number	Working as expected	PASS	MRITHULLA
3	Functional	OTP verification	Verify user otp using mail	1.Enter gmail id and enter password 2.click submit	Username: railways password: admin	OTP verified is to be displayed	Working as expected	FAIL	JESLENE
4	Functional	Login page	Verify user is able to log into application with InValid credentials	1.Enter into log in page 2.Click on My Account dropdown button 3.Enter inValid username/email in Email text box 4.Enter valid password in password text box	Username: railways password: admin	Application should show 'incorrect email or password ' validation message.	Working as expected	FAIL	ABINAYA
5	Functional	Display Train details	The user can view about the available train details	1.As a user, I can enter the start and destination to get the list of trains available connecting the above	Username: railways password: admin	A user can view about the available trains to enter start and destination details	Working as expected	PASS	VAISHNAVI

Shopenzer Testcases | Testscearnios

Test case 2

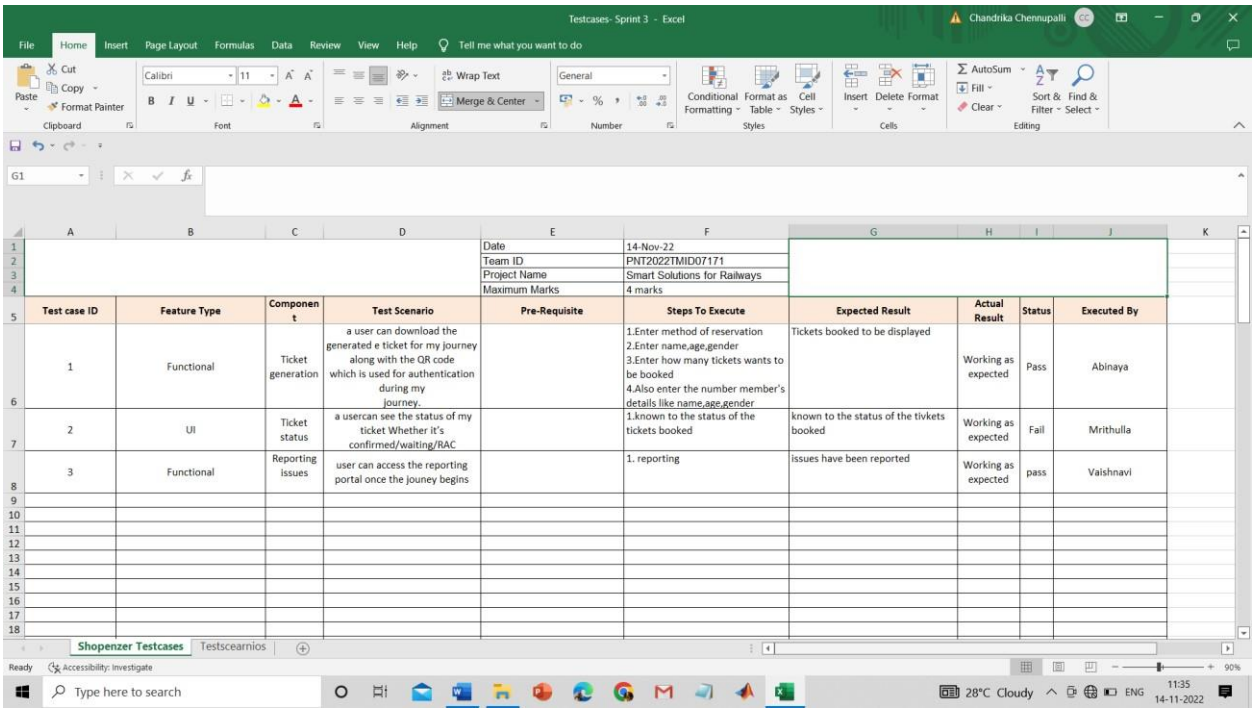


Testcases- Sprint 2 - Excel

Date	14-Nov-22	
Team ID	PNT2022TMID07171	
Project Name	Smart Solutions for Railways	
Maximum Marks	4 marks	

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Expected Result	Actual Result	Status	Executed By
1	Functional	Booking	user can provide the basic details such as a name, number, etc		1. Enter the member's details like name, number.	Tickets booked to be displayed	Working as expected	Pass	Abinaya
2	UI	Booking seats	User can choose the train, starting and ending destination, date of travel.		1. Known to which train is available	known to which the seats are available	Working as expected	fail	Jeslene
3	Functional	Payment	user, I can choose to pay through credit Card/debit card/UPI.		1.user can choose payment method 2.payment method	payment for the booked tickets to be done using payment method through either the following methods credit Card/debit	Working as expected	Fail	Mrithulla
4	Functional	Redirection	user can be redirected to the selected		1.After payment the user will be redirected to the previous page	After payment the user will be redirected to the previous page	Working as expected	pass	Vaishnavi

Test case 3

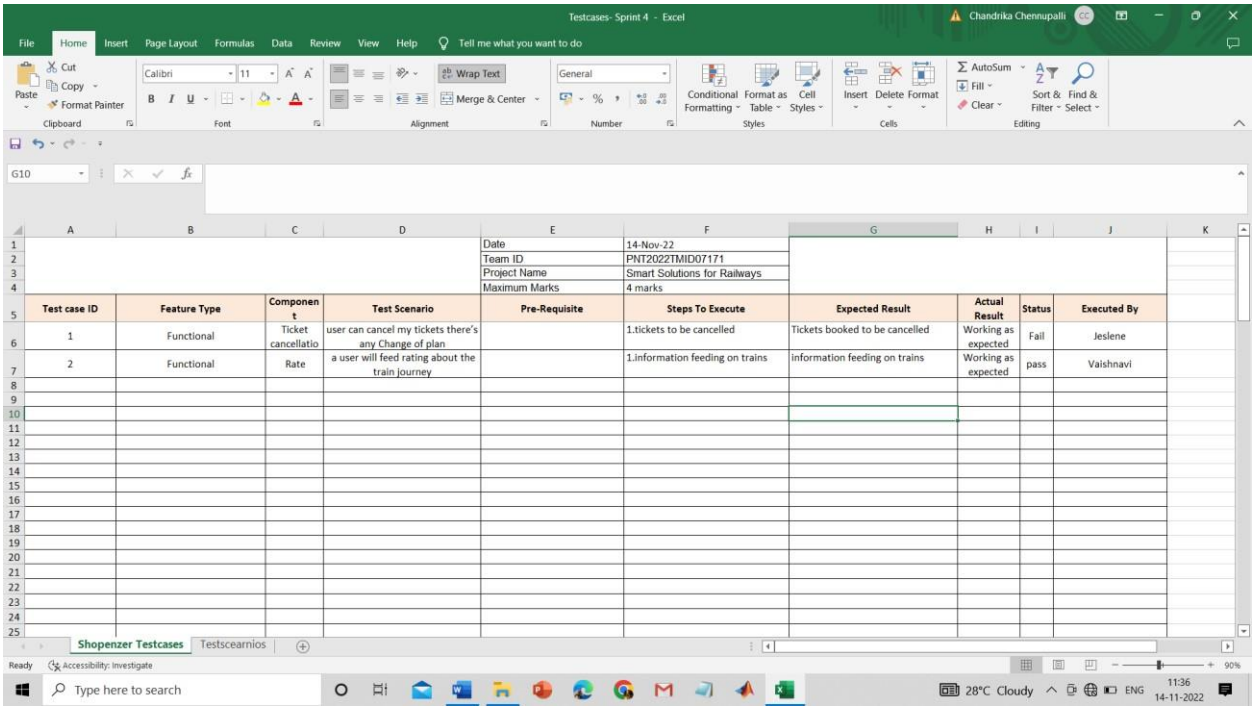


Testcases- Sprint 3 - Excel

Date	14-Nov-22	
Team ID	PNT2022TMID07171	
Project Name	Smart Solutions for Railways	
Maximum Marks	4 marks	

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Expected Result	Actual Result	Status	Executed By
1	Functional	Ticket generation	a user can download the generated e ticket for my journey along with the QR code which is used for authentication during my journey.		1.Enter method of reservation 2.Enter name,age,gender 3.Enter how many tickets wants to be booked 4.Also enter the number member's details like name,age,gender	Tickets booked to be displayed	Working as expected	Pass	Abinaya
2	UI	Ticket status	a usercan see the status of my ticket Whether it's confirmed/waiting/RAC		1.known to the status of the tickets booked	known to the status of the tivkets booked	Working as expected	Fail	Mrithulla
3	Functional	Reporting issues	user can access the reporting portal once the jouney begins		1. reporting	issues have been reported	Working as expected	pass	Vaishnavi

Test case 4



				Date	14-Nov-22				
				Team ID	PNT2022TMID07171				
				Project Name	Smart Solutions for Railways				
				Maximum Marks	4 marks				
Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Expected Result	Actual Result	Status	Executed By
1	Functional	Ticket cancellatio	user can cancel my tickets there's any Change of plan		1.tickets to be cancelled	Tickets booked to be cancelled	Working as expected	Fail	Jeslene
2	Functional	Rate	a user will feed rating about the train journey		1.information feeding on trains	information feeding on trains	Working as expected	pass	Vaishnavi

9. ADVANTAGES

- The passengers can use this application, while they are travelling alone to ensure their safety.
- It is easy to use.
- It has minimized error rate.

10. DISADVANTAGES

 Network issues may arise.

11. CONCLUSION

Almost all the countries across the globe strive to meet the demand for safe, fast, and reliable rail services. Lack of operational efficiency and reliability, safety, and security issues, besides aging railway systems and practices are haunting various countries to bring about a change in their existing rail infrastructure. The global rail industry struggles to meet the increasing demand for freight and passenger transportation due to lack of optimized use of rail network and inefficient use of rail assets. Often, they suffer from the lack in smart technologies and latest technological updates to provide the most efficient passenger services. This is expected to induce rail executives to build rail systems that are smarter and more efficient. The passenger reservation system of Indian Railways is one of the world's largest reservation models. Daily about one million passengers travel in reserved accommodation with Indian Railways. Another sixteen million travel with unreserved tickets in Indian Railways. In this vast system, it is a herculean task to efficiently handle the passenger data, which is a key point of consideration now-a-days. But the implementation of the latest technological updates in this system gradually turns inevitable due to increasing demand for providing the most efficient passenger services. Handling the passenger data efficiently backed by intelligent processing and timely retrieval would help backing up the security breaches. Here we've explored different issues of implementing smart computing in railway systems pertaining to reservation models besides pointing out some future scopes of advancement. Most significant improvements have been evidenced by more informative and user-friendly websites, mobile applications for real-time information about vehicles in motion, and e-ticket purchases and timetable information implemented at stations and stops. With the rise of Industry, railway companies can now ensure that they are prepared to avoid the surprise of equipment downtime. Like above mentioned, the developed application of our project can lead the passenger who travel can travel safely without any fear.

12. FUTURE SCOPE

This application is ensured for safety for the passengers while they are travelling alone as well as they travel with their family or friends.

In future, this application may also be used by passengers who travel through bus. By further enhancement of the application the passengers can explore more features regarding their safety.

13. APPENDIX

13.1 Source Code

FUNCTIONS IN NODE RED

FUNCTION NODE COMMAND TO INDICATE THE AVAILABLE SEATS:

```
var a=global.get('a') var s= [] for(let
i=0;i<a.length==0;i++){ s.push(a[i])
}
if(s.length==0){
  msg.options=[{"No seats available":0}]
}
else{
msg.options= s
}
msg.payload= s
return msg;
```

FUNCTION NODE COMMAND TO CHOOSE THE AVAILABLE SEATS:

```
var s=global.get('s') var a=global.get('a')
function reg(x){
for(let i=0;i<a.length;i++){ if(a[i]==x){
        a.splice(i,1)
      }
  }
} if(s==1){ global.set('s1',s)
reg(s)
}
else if(s==2){ global.set('s2',s)
reg(s)
}
else if(s==3){
global.set('s3',s) reg(s)
```

```
}
else                    if(s==4){
global.set('s4',s)      reg(s)
}
else                    if(s==4){
global.set('s4',s)      reg(s)
}
return msg;
```

FUNCTION NODE COMMAND TO STORE DATA IN DATABASE:

```
var m=global.get('m')      var d=new Date();        var
utc=d.getTime()+(d.getTimezoneOffset()*60000);        var
offset=5.5;
newDate=new    Date(utc+(3600000*offset));       var
n=newDate.toISOString() var date=n.slice(0,10)
var time=n.slice(11,19)
var d1=date+','+time msg.payload={
   "_id":d1,
   "Name":m.Name,
"Age":m.Age,
   "Mobile":m.Num,
   "boarding":global.get('b'),
   "destination":global.get('d'),
   "Seat":global.get('s')
}
```

return msg();

PYTHON SCRIPT TO SCAN QR CODE:

```
   import cv2 import numpy as np import time import pyzbar.pyzbar

as       pyzbar       from     pyzbar.pyzbar import decode    from
 ibmcloudant.cloudant_v1 import CloudantV1 from ibmcloudant
import CouchDbSessionAuthenticator                from
 ibm_cloud_sdk_core.authenticators import BasicAuthenticator


 authenticator  =  BasicAuthenticator('apikey-v2-125rwcp4ifi6zz2ly1cq0kakyjn98du2ysgc72h53lzi',
 'af693938842290ec2c254461754447b5') service =
```

```
CloudantV1(authenticator=authenticator)


service.set_service_url('https://apikey-v2-
125rwcp4ifi6zz2ly1cq0kakyjn98du2ysgc72h53lzi:af693938842290ec2c254461754447b5@82d874
9943954f 46-a190-6a186bee5051-bluemix.cloudantnosqldb.appdomain.cloud')


cap= cv2.VideoCapture(0) font = cv2.FONT_HERSHEY_PLAIN

while

True:
  _, frame = cap.read()  decodedObjects = pyzbar.decode(frame)  for obj in

decodedObjects:  #print ("Data", obj.data)  a=obj.data.decode('UTF-8')

cv2.putText(frame, "Ticket", (50,

50), font, 2,  (255, 0, 0), 3)
   #print (a)
    try:
      response = service.get_document(db='booking',doc_id = a).get_result()  print(response)

time.sleep(5)  except Exception as e:  print("NOT A VALID TICKER")  time.sleep(5)



  cv2.imshow("Frame",frame)  if cv2.waitKey(1)
& 0xFF ==ord('q'):
    break


cap.release() cv2.destroyAllWindows()
client.disconnect()
```

PYTHON CODE FOR GPS:

```
import wiotp.sdk.device import time import random myConfig = {

"identity": {
        "orgId": "dks66l",
        "typeId": "Sudha",
        "deviceId":"45"
 },
 "auth": {
        "token": "sudha2002@"
 }
   }
def myCommandCallback (cmd):    print ("Message received from IBM IoT Platform:
 %s" % cmd.data['command'])      m=cmd.data['command']



    client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None) client.connect()


    def pub (data):
        client.publishEvent(eventId="status",  msgFormat="json",  data=myData,  qos=0,  onPublish=None)

        print ("Published data Successfully: %s", myData)



    while True:
  myData={'name': 'Train1', 'lat':13.08363 , 'lon': 80.27080}
   pub (myData)          time.sleep (2)
   myData={'name': 'Train2', 'lat': 12.40797, 'lon': 79.81410}          pub  (myData)

   time.sleep (2)

myData={'name': 'Train1', 'lat': 11.83331, 'lon': 79.37465}
        pub(myData)          time.sleep(6)

   myData={'name': 'Train1', 'lat': 11.59664, 'lon': 78.69899}          pub  (myData)

   time.sleep (6)
```

```
myData={'name': 'Train1', 'lat': 11.63431, 'lon': 78.11122}
        pub (myData)
time.sleep  (6)        myData={'name':  'Train1',  'lat':

11.32207, 'lon': 77.61684}                pub    (myData)

        time.sleep (6)

myData={'name':  'Train1',  'lat':  11.03107,  'lon':  76.96864}          pub  (myData)

time.sleep (6)            client.commandCallback = myCommandCallback

            client.disconnect ()
```

13.2 GitHub

GitHub link:

https://github.com/IBM-EPBL/IBM-Project-12299-1659446880